

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2002年11月14日

出 願 番 号

Application Number:

特願2002-330323

[ST.10/C]:

[JP2002-330323]

出 願 人

Applicant(s):

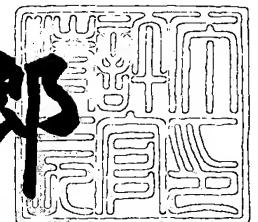
株式会社日立製作所

U.S. Appln. Filed 6-27-03
Inventor: K. Serizawa et al
Mattingly Stanger & Malor
Docket NIT-377

2003年 5月27日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田信一郎



出証番号 出証特2003-3039863

【書類名】 特許願

【整理番号】 NT02P0731

【提出日】 平成14年11月14日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 12/02

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所 システム開発研究所内

【氏名】 芹沢 一

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社日立製作所

【代理人】

【識別番号】 100068504

【弁理士】

【氏名又は名称】 小川 勝男

【電話番号】 03-3661-0071

【選任した代理人】

【識別番号】 100086656

【弁理士】

【氏名又は名称】 田中 恭助

【電話番号】 03-3661-0071

【選任した代理人】

【識別番号】 100094352

【弁理士】

【氏名又は名称】 佐々木 孝

【電話番号】 03-3661-0071

【手数料の表示】

【予納台帳番号】 081423

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 仮想ボリュームの記憶領域割当方法、その装置及びプログラム

【特許請求の範囲】

【請求項 1】

仮想ボリュームに対して少なくとも 1 台のストレージ装置の記憶領域のうちの空きの記憶領域を割り当てる方法において、

要求サイズに対して残りサイズが指定された領域サイズ上限値未満となるまで前記空きの記憶領域から記憶領域を切り出して割り当て、前記残りサイズが前記領域サイズ上限値未満となったとき、前記残りサイズについて前記残りサイズ以上の最小の 2 のべき乗数のサイズの記憶領域を前記空きの記憶領域から切り出して割り当てることを特徴とする仮想ボリュームの記憶領域割当方法。

【請求項 2】

前記空きの記憶領域が複数の連続空き領域から成るとき、最大サイズをもつ前記連続空き領域を割当対象として選択することを特徴とする請求項 1 記載の仮想ボリュームの記憶領域割当方法。

【請求項 3】

前記空きの記憶領域の両端に隣接する記憶領域が割当済である場合に、解放しにくい方の割当済領域に隣接するように記憶領域を切り出して割り当てることを特徴とする請求項 1 記載の仮想ボリュームの記憶領域割当方法。

【請求項 4】

前記空きの記憶領域が互いに負荷分散可能な複数の空き領域のグループに区分される場合に、前記仮想ボリュームを負荷分散可能な複数の領域に分割して割り当てるよう要求されたとき、指定された分割数に基づいて要求サイズを前記グループに分割して割り当てることを特徴とする請求項 1 記載の仮想ボリュームの記憶領域割当方法。

【請求項 5】

前記グループの数が前記分割数より多いときに、前記グループのうち前記分割数個のグループを空き領域の合計が大きい順に選び、選んだグループに要求されたサイズを分割して割り当てることを特徴とする請求項 4 記載の仮想ボリューム

の記憶領域割当方法。

【請求項 6】

前記残りサイズの割当の前に、前記空きの記憶領域から前記領域サイズ上限値の整数倍であって前記要求サイズを越えない最大の記憶領域を割り当てることを特徴とする請求項 1 記載の仮想ボリュームの記憶領域割当方法。

【請求項 7】

前記残りサイズの割当の前に、前記空きの記憶領域から前記要求サイズを越えない最大の 2 のべき乗数のサイズをもつ記憶領域を割り当てることを特徴とする請求項 1 記載の仮想ボリュームの記憶領域割当方法。

【請求項 8】

仮想ボリューム上の記憶領域のアドレスとストレージ装置内の論理ユニットのアドレス及び前記論理ユニット内記憶領域のアドレスとの対応情報を格納するアクセス変換テーブル手段と、前記アクセス変換テーブル手段を参照して前記仮想ボリュームに対する入出力要求を前記ストレージ装置内の記憶領域に対する入出力要求に変換する手段とを有する仮想化装置において、

前記仮想ボリュームに対して前記ストレージ装置の記憶領域のうちの空きの記憶領域を割り当てる要求を受け付ける手段と、要求サイズに対して残りサイズが指定された領域サイズ上限値未満となるまで前記空きの記憶領域から記憶領域を切り出して割り当てる手段と、前記残りサイズが前記領域サイズ上限値未満となったとき、前記残りサイズについて前記残りサイズ以上の最小の 2 のべき乗数のサイズの記憶領域を前記空きの記憶領域から切り出して割り当てる手段と、割当要求に対して記憶領域の割当が済んだ後に、割当結果に基づいて前記アクセス変換テーブル手段の内容を更新する手段とを有することを特徴とする仮想化装置。

【請求項 9】

さらに前記空きの記憶領域が複数の連続空き領域から成るとき、最大サイズをもつ前記連続空き領域を割当対象として選択する手段を有することを特徴とする請求項 8 記載の仮想化装置。

【請求項 10】

前記空きの記憶領域の両端に隣接する記憶領域が割当済である場合に、解放し

にくい方の割当済領域に隣接するように記憶領域を切り出して割り当てる手段を有することを特徴とする請求項 8 記載の仮想化装置。

【請求項 1 1】

前記空きの記憶領域が互いに負荷分散可能な複数の空き領域のグループに区分される場合に、前記仮想ボリュームを負荷分散可能な複数の領域に分割して割り当てるよう要求されたとき、指定された分割数に基づいて要求サイズを前記グループに分割して割り当てる手段を有することを特徴とする請求項 8 記載の仮想化装置。

【請求項 1 2】

前記グループの数が前記分割数より多いときに、前記グループのうち前記分割数個のグループを空き領域の合計が大きい順に選び、選んだグループに要求されたサイズを分割して割り当てる手段を有することを特徴とする請求項 1 1 記載の仮想化装置。

【請求項 1 3】

請求項 8 に記載の仮想化装置を内蔵することを特徴とするストレージ装置。

【請求項 1 4】

仮想化装置またはホストプロセッサに、仮想ボリュームに対して少なくとも 1 台のストレージ装置の記憶領域のうちの空きの記憶領域を割り当てる機能を実現させるためのプログラムにおいて、

前記仮想化装置またはホストプロセッサに、要求サイズに対して残りサイズが指定された領域サイズ上限値未満となるまで前記空きの記憶領域から記憶領域を切り出して割り当てる機能、および前記残りサイズが前記領域サイズ上限値未満となったとき、前記残りサイズについて前記残りサイズ以上の最小の 2 のべき乗数のサイズの記憶領域を前記空きの記憶領域から切り出して割り当てる機能を実現させるためのプログラム。

【請求項 1 5】

さらに前記空きの記憶領域が複数の連続空き領域から成るとき、最大サイズをもつ前記連続空き領域を割当対象として選択する機能を実現させるための請求項 1 4 記載のプログラム。

【請求項 1 6】

前記空きの記憶領域の両端に隣接する記憶領域が割当済である場合に、解放しにくい方の割当済領域に隣接するように記憶領域を切り出して割り当てる機能を実現させるための請求項 1 4 記載のプログラム。

【請求項 1 7】

前記空きの記憶領域が互いに負荷分散可能な複数の空き領域のグループに区分される場合に、前記仮想ボリュームを負荷分散可能な複数の領域に分割して割り当てるよう要求されたとき、指定された分割数に基づいて要求サイズを前記グループに分割して割り当てる機能を実現させるための請求項 1 4 記載のプログラム。

【請求項 1 8】

前記グループの数が前記分割数より多いときに、前記グループのうち前記分割数個のグループを空き領域の合計が大きい順に選び、選んだグループに要求されたサイズを分割して割り当てる機能を実現させるための請求項 1 7 記載のプログラム。

【請求項 1 9】

実記憶領域を保持する少なくとも 1 台のストレージ装置と、前記ストレージ装置の前記実記憶領域に対するデータ読み書きを起動する少なくとも 1 台のホストプロセッサと、前記ホストプロセッサと前記ストレージ装置との間に介在し前記ホストプロセッサに対して仮想ボリュームを提供する仮想化装置と、前記仮想化装置に対して記憶領域の割当要求を発行する管理コンソールとを有するシステムにおいて、

前記仮想化装置は、前記仮想ボリューム上の記憶領域のアドレスと前記ストレージ装置内の論理ユニットのアドレス及び前記論理ユニット内記憶領域のアドレスとの対応情報を格納するアクセス変換テーブル手段と、前記アクセス変換テーブル手段を参照して前記ホストプロセッサから受信した前記仮想ボリュームに対する入出力要求を前記ストレージ装置内の記憶領域に対する入出力要求に変換する手段と、前記仮想ボリュームに対して前記ストレージ装置の記憶領域のうちの空きの記憶領域を割り当てる前記管理コンソールからの要求を受け付ける手段と

、要求サイズに対して残りサイズが指定された領域サイズ上限値未満となるまで前記空きの記憶領域から記憶領域を切り出して割り当てる手段と、前記残りサイズが前記領域サイズ上限値未満となったとき、前記残りサイズについて前記残りサイズ以上の最小の2のべき乗数のサイズの記憶領域を前記空きの記憶領域から切り出して割り当てる手段と、割当要求に対して記憶領域の割当が済んだ後に、割当結果に基づいて前記アクセス変換テーブル手段の内容を更新する手段とを有することを特徴とするシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、ストレージ仮想化システムにおいて仮想ボリュームに対する実ストレージ領域の割り当て技術に関する。

【0002】

【従来の技術】

ホスト計算機とネットワークで接続されたストレージ装置を仮想化してホスト計算機に見せる技術は、例えば特開2000-339098号公報（特許文献1）、米国特許第6,216,202号明細書（特許文献2）などに記載されている。

【0003】

また特開2001-350707号公報（特許文献3）には、実ストレージ又はその部分領域の性能の評価値とシステム管理者が指定する性能要求値とを比較し、その結果に基づいてストレージの領域をホスト計算機に割り当てる技術が示されている。

【0004】

さらに米国特許第6,131,150号明細書（特許文献4）は、b u d d y a l g o r i t h m （以下 B. A. と略記する）と呼ばれる技術を開示する。B. A. は、メモリの空き領域の断片化（フラグメンテーション）を防止するための方法である。メモリ領域を2のべき乗数のサイズを単位として割り当てるよう管理することによって、メモリ解放時に隣接の空き領域と再統合すること

ができるのでフラグメンテーションを防止できる。

【0005】

【特許文献1】

特開2000-339098号公報

【特許文献2】

米国特許第6, 216, 202号明細書

【特許文献3】

特開2001-350707号公報

【特許文献4】

米国特許第6, 131, 150号明細書

【0006】

【発明が解決しようとする課題】

ストレージの仮想化により、実ストレージ装置内の記憶領域（以下実領域と呼ぶ）の割当てと解放が頻繁に行われると、ストレージ装置内での空き領域が断片化する。その結果、一つの仮想ボリュームに割り当てる実領域の数が増加する。実領域の数の増加は、管理コストの増大や転送性能の低下、信頼性の低下の原因となる。特許文献1, 2及び3に開示される仮想ボリュームの構成方法はフラグメンテーションの回避を考慮していない。第1の本発明が解決しようとする課題は、フラグメンテーションの回避である。

【0007】

一方、特許文献4に開示されるB. A. は、割り当てる領域のサイズが大きくなると効率が低下する。これは割当要求された領域のサイズとこのサイズを越える最小の2のべき乗数のサイズとの差が無駄になるためである。ストレージの実領域割当では割り当てるサイズが比較的大きい場合があるので、この効率低下が問題になることがある。第2の本発明が解決しようとする課題は、ストレージの実記憶領域割当の効率低下の回避である。

【0008】

またストレージの仮想化システムでは実領域の割当ては、領域解放と比較しその頻度が多い。実領域の割当は仮想ボリュームの容量追加のたびに行われるが、

解放は仮想ボリューム全体を解放するときに限られるためである。従って、連続の空き領域を増加させるためには、解放の時に領域が連続して解放される可能性が大きくなるように、割り当てる実領域を選択する必要がある。第3の本発明が解決しようとする課題は、連続して解放されるような実領域を選択することである。

【0009】

また仮想ボリュームに割り当てる実領域が複数ある場合は、それらの実領域は性能や信頼性が同等で、かつ負荷の集中を回避するために複数のストレージ装置に分散するように負荷分散を考慮したストレージ装置の選択を行うことが望ましい。さらに作成した仮想ボリュームに後から実領域を追加するときも、負荷分散を考慮したストレージ装置の選択が望ましい。第4の本発明が解決しようとする課題は、負荷分散を考慮した実領域の選択である。

【0010】

本発明の目的は、上記課題のうちの少なくとも1つを解決する仮想ボリュームの記憶領域割当技術を提供することにある。

【0011】

【課題を解決するための手段】

少なくとも上記第2の課題を解決する本発明は、仮想ボリュームに対して少なくとも1台のストレージ装置の記憶領域のうちの空きの記憶領域を割り当てる技術において、要求サイズに対して残りサイズが指定された領域サイズ上限値未満となるまで空きの記憶領域から記憶領域を切り出して割り当て、残りサイズが領域サイズ上限値未満となったとき、この残りサイズについて残りサイズ以上の最小の2のべき乗数のサイズの記憶領域を空きの記憶領域から切り出して割り当てる仮想ボリュームの記憶領域割当技術を特徴とする。さらに本発明は、上記第1の課題、第3の課題または第4の課題を解決するように構成される。

【0012】

【発明の実施の形態】

以下、本発明の実施形態について図面を用いて説明する。

【0013】

図 1 は、本発明に係るシステム全体の構成を示す図である。システムは、少なくとも 1 台のホストプロセッサ 1 2、少なくとも 1 台のストレージ装置 1 3、仮想化スイッチ 1 1 及び管理コンソール 1 4 から構成される。仮想化スイッチ 1 1 は、図示するように伝送路によって他の装置と接続されており、他の装置と通信することができる。ホストプロセッサ 1 2 は、ストレージ装置に対するデータ読み書きを起動する計算機である。

【 0 0 1 4 】

ストレージ装置 1 3 は、それ自身ストレージ制御部を備えたストレージサブシステムであり、少なくとも 1 つの LU (L o g i c a l U n i t) 1 3 1 を有する。LU 1 3 1 は、ストレージ装置 1 3 内の記憶装置上に存在する少なくとも 1 つの実領域 1 3 2 から構成されており、ホストプロセッサ 1 2 など外部に対して論理的に独立した 1 つのストレージ装置を提供するものである。LU 1 3 1 は、論理ユニットであるが、通常 R A I D グループなど物理的装置に対応する。実領域 1 3 2 は、LU 1 3 1 内の記憶領域を任意に区分する論理領域であり、連続したアドレスをもつ領域である。実領域 1 3 2 内の任意領域は、LU アドレスと実領域 1 3 2 内アドレスによってアクセスされる。

【 0 0 1 5 】

仮想化スイッチ 1 1 とホストプロセッサ 1 2 との間および仮想化スイッチ 1 1 とストレージ装置 1 3 との間は、ファイバー・チャネルのような伝送路によって接続される。仮想化スイッチ 1 1 は、ホストプロセッサ 1 2 とストレージ装置 1 3 との間の伝送路に介入し、ホストプロセッサ 1 2 側の通信経路をストレージ装置 1 3 側の通信経路にスイッチする機能を有する。また仮想化スイッチ 1 1 は、ホストプロセッサ 1 2 に対して仮想ボリューム 1 0 0 を提供する。仮想ボリューム 1 0 0 は、少なくとも 1 つの LU 1 3 1 に属する実領域 1 3 2 を 1 つ以上集めて構成される仮想化された記憶装置である。仮想ボリューム 1 0 0 内の記憶領域全体には連続したアドレスが付されている。ホストプロセッサ 1 2 は、ストレージ装置 1 3 の LU 1 3 1 に直接アクセスする代わりに、仮想ボリューム 1 0 0 のアドレス及び仮想ボリューム 1 0 0 内記憶領域のアドレスを指定して仮想ボリューム 1 0 0 のデータを読み書きすることができる。

【 0 0 1 6 】

管理コンソール 1 4 は、仮想ボリューム 1 0 0 を新規作成したりその記憶領域の構成を変更するためにシステム管理者によって使用される入出力装置であり、表示装置と入力装置を備える。

【 0 0 1 7 】

図 2 は、仮想化スイッチ 1 1 の内部構成を示す図である。仮想化スイッチ 1 1 は、入力ポート 2 4 0、出力ポート 2 5 0、転送部 2 3 0、処理部 2 1 0 および記憶部 2 2 0 から構成される。入力ポート 2 4 0 は、仮想化スイッチ 1 1 がホストプロセッサ 1 2 と通信するためのポートである。出力ポート 2 5 0 は、仮想化スイッチ 1 1 がストレージ装置 1 3 と通信するためのポートである。転送部 2 3 0 は、そのメモリ上に転送情報テーブル 2 3 1 を保持する。転送情報テーブル 2 3 1 は、各入力ポート 2 4 0 と通信可能なホストプロセッサ 1 2、各出力ポート 2 5 0 と通信可能なストレージ装置 1 3 の対応情報を格納する。転送部 2 3 0 は、転送情報テーブル 2 3 1 を参照し、入力ポート 2 4 0 がホストプロセッサ 1 2 から受信した入出力要求を要求先のストレージ装置 1 3 が接続される出力ポート 2 5 0 へ転送する。また出力ポート 2 5 0 がストレージ装置 1 3 から受信した応答情報やデータを該当するホストプロセッサ 1 2 が接続される入力ポート 2 4 0 へ転送する。ただしホストプロセッサ 1 2 から受け取った入出力要求のうち仮想ボリューム 1 0 0 に対する入出力要求については、後述のアクセス変換部 2 1 2 の処理が介入し、その結果として決定されたストレージ装置 1 3 へ入出力要求を送信する。処理部 2 1 0 は、プロセッサとメモリとを備え、このプロセッサがメモリに格納されたプログラムを実行することによって以下に詳述する仮想化スイッチ 1 1 の機能を実現する。記憶部 2 2 0 は、処理部 2 1 0 が参照する管理テーブルやデータを格納する。

【 0 0 1 8 】

図 3 は、処理部 2 1 0 および記憶部 2 2 0 の内部構成を示す図である。処理部 2 1 0 は、そのメモリ上に仮想ボリューム定義部 2 1 1、アクセス変換部 2 1 2、割当処理部 2 1 3、実領域切出部 2 1 4 およびプール作成部 2 1 5 の各プログラムを格納し、そのプロセッサによってこれらプログラムを実行する。記憶部 2

20は、仮想ボリューム管理テーブル221、実領域管理テーブル222、プール管理テーブル223、アクセス変換テーブル224および領域サイズ上限値225を記憶する。

【0019】

アクセス変換テーブル224は、仮想ボリューム100ごとに存在し、仮想ボリューム100上の記憶領域のアドレスとストレージ装置13内のLU131のアドレス及び実領域132内記憶領域のアドレスとの対応情報を保持するテーブルである。アクセス変換テーブル224は、仮想ボリューム100内の記憶領域の構成を変更したとき更新される。

【0020】

アクセス変換部212は、アクセス変換テーブル224を参照してホストプロセッサ12から受け取った仮想ボリューム100に対する入出力要求を対応するストレージ装置13およびLU131への入出力要求に変換する。なお各入力ポート240ごとに分散してアクセス変換テーブル224およびアクセス変換部212をもつと性能向上できる。

【0021】

実領域管理テーブル222は、LU131ごとに存在し、そのLU131に含まれる実領域132を管理するために使用されるテーブルである。各実領域管理テーブル222は、ストレージ装置ID321、LU-ID322、グループID323および実領域リスト324を格納する。ストレージ装置ID321は、当該LUを保持するストレージ装置13の識別子である。LU-ID322は当該LUの識別子である。グループID323は、ストレージ装置13側の複数のLU131を性能によってグループ化するとき当該LUが属するグループの識別子である。

【0022】

ここで言うグループは、複数のストレージ装置13あるいはLU131にかかるアクセス負荷を分散するときの記憶単位となるものである。例えばストレージ装置13が保持するLU131ごとあるいは実領域132ごとに負荷を分散できるなら、各LU131ごとあるいは負荷分散可能な実領域132ごとにグループ

を区分してよい。反対にストレージ装置 1 3 が負荷分散機能をもたない場合には、そのストレージ装置 1 3 が保持する LU 1 3 1 はすべて同一グループとなる。すなわち同一のグループに属する記憶装置あるいは記憶領域は、負荷分散のためにはそれ以上分割できない単位となる。負荷分散を要求する仮想ボリュームに対しては、1 つの仮想ボリュームについて複数のグループが割り当てられる。

【 0 0 2 3 】

実領域リスト 3 2 4 は、少なくとも 1 つのエントリ 3 2 5 から構成される。各エントリ 3 2 5 は、実領域 1 3 2 に対応して設けられ、領域 ID、サイズ、仮想ボリューム識別子及び実領域の解放し易さの各項目から成る。領域 ID は当該実領域 1 3 2 を特定するための識別子、サイズは当該実領域 1 3 2 のサイズである。仮想ボリューム識別子は、当該実領域 1 3 2 が割り当てられている仮想ボリュームの識別子である。実領域の解放のし易さは、その実領域が仮想ボリューム 1 0 0 に割り当てられた後にその実領域が解放し易いか否かのレベルあるいは優先度を示す情報である。その実領域 1 3 2 のデータを他の実領域 1 3 2 に移動する可能性が高い場合には解放し易い実領域 1 3 2 と言える。逆に移動する可能性が低い場合には解放し難い実領域 1 3 2 と言える。実領域リスト 3 2 4 内のエントリ 3 2 5 は、実領域 1 3 2 のアドレス順に配列されている。グループ内の実領域 1 3 2 のアドレスは連続しており、エントリ 3 2 5 はそのグループのすべての実領域を網羅する。実領域 1 3 2 が空きであるようなエントリは、その仮想ボリューム識別子が空となっている。

【 0 0 2 4 】

仮想ボリューム管理テーブル 2 2 1 は、仮想ボリューム 1 0 0 ごとに存在し、識別子 3 1 1、分割数 3 1 2、要求性能 3 1 3、割当てサイズ 3 1 4、割当済実領域リスト 3 1 5 及び仮割当実領域リスト 3 1 6 を格納する。識別子 3 1 1 は、当該仮想ボリューム 1 0 0 の識別子である。分割数 3 1 2 は、当該ボリューム 1 0 0 を複数の領域に分割し、各々の領域へのアクセスを負荷分散する場合にその分散する領域の数である。要求性能 3 1 3 は、システム管理者が当該仮想ボリューム 1 0 0 に対して要求する性能と信頼性を示す数値である。性能はデータ転送速度によって表現される。また信頼性はランク付けされた信頼度である。割当て

サイズ 3 1 4 は、割当処理において当該仮想ボリューム 1 0 0 に割り当てる実領域のサイズである。特許文献 3 には性能指標についての記載がある。割当済実領域リスト 3 1 5 は、当該仮想ボリューム 1 0 0 に割当済の実領域 1 3 2 の実領域 ID 3 1 7 のリストである。仮割当実領域リスト 3 1 6 は、当該仮想ボリューム 1 0 0 に割り当てるために予約された実領域 1 3 2 の実領域 ID 3 1 7 のリストである。

【 0 0 2 5 】

プール管理テーブル 2 2 3 は、上記のグループごとに存在し、性能指標 3 5 1、グループ ID 3 2 3、空き領域の合計 3 5 3 および実領域リスト 3 5 4 を格納する。性能指標 3 5 1 は、当該プールを構成する実領域 1 3 2 の性能と信頼性を示す数値である。グループ ID 3 2 3 は、当該プールに対応するグループの識別子である。空き領域の合計 3 5 3 は、当該プールの空き領域の合計容量である。実領域リスト 3 5 4 は、当該プールを構成する空きの実領域 1 3 2 の識別子（実領域 ID）のリストである。

【 0 0 2 6 】

領域サイズ上限値 2 2 5 は、領域割当において許容し得る無駄になる実領域 1 3 2 のサイズの上限値を格納する。領域サイズ上限値 2 2 5 は、2 のべき乗数で設定される。領域サイズ上限値 2 2 5 は、システム管理者によって設定されるデータである。

【 0 0 2 7 】

仮想ボリューム定義部 2 1 1 は、管理コンソール 1 4 を介してシステム管理者から仮想ボリューム 1 0 0 の割当要求と今回割り当てる実領域について、割当てサイズ 3 1 4、分割数 3 1 2 および要求性能 3 1 3 を受け取り、仮想ボリューム管理テーブル 2 2 1 を新規作成または更新する。仮想ボリューム 1 0 0 の記憶容量を増やすときは仮想ボリューム管理テーブル 2 2 1 の更新となる。次に仮想ボリューム定義部 2 1 1 は、プール作成部 2 1 5 にプールの作成を依頼する。プール作成部 2 1 5 は、すべての実領域管理テーブル 2 2 2 について空きの実領域 1 3 2 を探し、グループごとにグループ ID 3 2 3 を作成し、各グループの空きの実領域 1 3 2 の識別子の集合を実領域リスト 3 5 4 に格納し、空き容量の合計 3

5 3 を設定する。またそのグループについてあらかじめ定義された性能指標 3 5 1 を設定する。すべてのプール管理テーブル 2 2 3 が作成された後、仮想ボリューム定義部 2 1 1 は、それらのプール群と、作成または更新された仮想ボリューム管理テーブル 2 2 1 を割当処理部 2 1 3 に渡し、実領域 1 3 2 の割当て処理を依頼する。

【 0 0 2 8 】

図 4 は、割当処理部 2 1 3 の処理の流れを示すフローチャートである。まず割当処理部 2 1 3 は、当該仮想ボリューム管理テーブル 2 2 1 の割当済実領域リスト 3 1 5 と実領域管理テーブル 2 2 2 を参照して、当該仮想ボリューム 1 0 0 が割り当てられている実領域 1 3 2 のグループのグループ ID 3 2 3 を列挙する（ステップ 2 0 0 1）。仮想ボリューム 1 0 0 を新規に作成する場合には、列挙すべきグループ ID 3 2 3 がないので、何もしない。次に割当処理部 2 1 3 は、変数 p 1 に分割数 3 1 2 から列挙したグループの数（新規割当の場合は 0）を差し引いた数を代入する（ステップ 2 0 0 2）。変数 p 1 は新たに探すプールの数を意味する。

【 0 0 2 9 】

変数 p 1 が 0 を超える場合（ステップ 2 0 0 3 Y）、割当処理部 2 1 3 は、ステップ 2 0 0 4 からステップ 2 0 0 6 までを実行して、実領域 1 3 2 を割当てるプールを選択する。すなわちまず割当処理部 2 1 3 は、要求仕様 3 1 3 とステップ 2 0 0 1 で列挙したグループ ID 3 2 3 以外をもつプール管理テーブル 2 2 3 の性能指標 3 5 1 とを比較して要求仕様 3 1 3 を満たすプールを選択する（ステップ 2 0 0 4）。特許文献 3 には要求仕様と性能指標との比較の方法についての記載がある。次に選択したプールの数と変数 p 1 とを比較する（ステップ 2 0 0 5）。プールの数 < p 1 の場合（ステップ 2 0 0 5 Y）には、プールの数が不足するのでステップ 2 0 1 3 へ行く。プールの数 \geq p 1 の場合（ステップ 2 0 0 5 N）には空き容量の合計 3 5 3 が大きな順に p 1 個のプールを選択し（ステップ 2 0 0 6）、ステップ 2 0 1 3 へ進む。ただし p 1 個のプールの空き容量の合計 3 5 3 をすべて加えた総容量が割当てサイズ 3 1 4 より小さい場合には、総容量が割当てサイズ 3 1 4 に達するまで、空き容量の合計 3 5 3 が大きい順にさらに

プールを選択する。

【 0 0 3 0 】

一方 p 1 が 0 を越えない場合（ステップ 2 0 0 3 N）、割当処理部 2 1 3 は、ステップ 2 0 0 7 からステップ 2 0 1 2 までの処理を実行する。まず割当処理部 2 1 3 は、ステップ 2 0 0 1 で列挙したグループ I D 3 2 3 を持つプール管理テーブル 3 2 3、すなわちプール群を選択する（ステップ 2 0 0 7）。次に割当処理部 2 1 3 は、列挙したグループ I D 3 2 3 をもつ実領域管理テーブル 2 2 2 の実領域リスト 3 2 4 を探索し、すでに当該仮想ボリューム 1 0 0 に割り当てられた実領域 1 3 2 に隣接する連続空き領域を選択する（ステップ 2 0 0 8）。割り当てるべき空き領域が複数ある場合は、割当てサイズ 3 1 4 を適当に分割して各々を領域割当の対象とする。次に割当処理部 2 1 3 は、割当サイズと選択した連続空き領域群を指定して実領域切出部 2 1 4 を呼び出し、実領域切り出し処理を行う（ステップ 2 0 0 9）。その後、切り出された領域のリストを当該仮想ボリューム管理テーブル 2 2 1 の仮割当実領域リスト 3 1 6 に追加し、仮割当実領域リスト 3 1 6 を更新する（ステップ 2 0 1 0）。次に割当処理部 2 1 3 は、割当てサイズ 3 1 4 からステップ 2 0 0 9 で割り当てた領域のサイズを差し引く（ステップ 2 0 1 1）。割当てサイズ 3 1 4 が 0 以下になれば（ステップ 2 0 1 2 N）、処理を終了する。

【 0 0 3 1 】

次に割当処理部 2 1 3 は、各プールに割り当てる領域のサイズを割り振る（ステップ 2 0 1 3）。割り振り方は等分でもよいし、そうでなくてもよい。対象となるプールは、ステップ 2 0 1 3 に来るまでに選択したプールである。次に割当処理部 2 1 3 は、各プールについて実領域管理テーブル 2 2 2 を参照し、ステップ 2 0 1 3 で割り振ったサイズを越える連続の空き領域群があるか否か判定する（ステップ 2 0 1 4）。空き領域がない場合には、最大の連続空き領域を選択する（ステップ 2 0 1 6）。空き領域がある場合は、その近傍の実領域 1 3 2 の解放のしやすさが最も小さい（最も解放され難い）空き領域群を選ぶ（ステップ 2 0 1 5）。空き領域に隣接する一方の実領域 1 3 2 が解放され難く、他方の実領域 1 3 2 が解放され易い場合に、今回割り当てる領域を前者に隣接するように

割り当てると、後者が解放されたときに統合される連続空き領域のサイズがより大きくなる。

【0032】

次に割当処理部 2 1 3 は、ステップ 2 0 1 5 またはステップ 2 0 1 6 で選択した実領域群 1 3 2 と割当てサイズを指定して実領域切出部 2 1 4 を呼び出し、実領域切り出し処理を行う（ステップ 2 0 1 7）。切り出された領域のリストを当該仮想ボリューム管理テーブル 2 2 1 の仮割当て実領域リスト 3 1 6 に追加し、仮割当実領域リスト 3 1 6 を更新する（ステップ 2 0 1 8）。次に割当処理部 2 1 3 は、割当てサイズ 3 1 4 からステップ 2 0 1 7 で割り当てた領域のサイズを差し引く（ステップ 2 0 1 9）。割当てサイズ 3 1 4 が 0 を越えていれば（ステップ 2 0 2 0 Y）、ステップ 2 0 1 3 に戻る。割当てサイズが 0 以下になれば（ステップ 2 0 2 0 N）、処理を終了する。

【0033】

上記割当処理部 2 1 3 の処理手順によれば、システム管理者が仮想ボリューム 1 0 0 の割当要求時に指定した割当てサイズ 3 1 4、分割数 3 1 2 及び要求性能 3 1 3 のうち、割当てサイズ 3 1 4 を満足させることが最優先される。次いでステップ 2 0 0 4、2 0 0 5 により要求性能 3 1 3 を満足させることが優先される。分割数 3 1 2 を満足させることは最後の優先となる。また上記割当処理部 2 1 3 の処理手順によれば、ステップ 2 0 1 5 の処理により、できるだけフラグメンテーションが起きないような実領域の割り当て方をする。

【0034】

図 5 は、第 1 の実施例において実領域切出部 2 1 4 の処理の流れを示すフローチャートである。まず実領域切出部 2 1 4 は、初期化处理として割り当てた実領域 1 3 2 を割当処理部 2 1 3 に返すためにメモリ上に設けた連絡領域であるリストを空に初期化し（ステップ 2 1 0 1）、変数 *s i z e* に渡されたサイズを格納する（ステップ 2 1 0 2）。次に実領域切出部 2 1 4 は、渡された連続空き空き領域群について、1 つの空き領域の下位と上位に隣接する割当済の実領域 1 3 2 の解放されにくさを判定し、解放されにくい実領域 1 3 2 に隣接する空き領域を選択する（ステップ 2 1 0 3）。

【 0 0 3 5 】

次に実領域切出部 2 1 4 は、ステップ 2 1 0 4 から 2 1 1 6 までを繰り返すことによってステップ 2 1 0 3 で選択した空き領域について順に実領域 1 3 2 を切り出していく。まず実領域切出部 2 1 4 は、選択した空き領域のサイズが `size` 以下の場合には（ステップ 2 1 0 4 N）、選択した空きの実領域 1 3 2 の全体を切り出すこととし、この実領域 1 3 2 とそのサイズをリストに登録する（ステップ 2 1 0 5）。ここで切り出すとは、実領域管理テーブル 2 2 2 の該当するエントリ 3 2 5 に仮想ボリュームの識別子を設定することである。以下切り出すとは、一般に実領域管理テーブル 2 2 2 の該当するエントリ 3 2 5 を割当済のエントリ 3 2 5 と空きのエントリ 3 2 5 とに分割し、必要な領域 ID、サイズ及び仮想ボリューム 1 0 0 の識別子を設定することである。次にステップ 2 1 0 3 と同様の処理によって渡された連続空き領域群について次の空き領域を選択する（ステップ 2 1 0 6）。次に `size` からステップ 2 1 0 5 で登録した実領域 1 3 2 のサイズを差し引く（ステップ 2 1 0 7）。`size` が 0 になった場合（ステップ 2 1 0 8 Y）には、ステップ 2 1 1 7 へ進む。それ以外の場合はステップ 2 1 0 4 に戻る。

【 0 0 3 6 】

選択した空き領域のサイズが `size` より大きい場合（ステップ 2 1 0 4 Y）には、ステップ 2 1 0 9 からステップ 2 1 1 6 までを実行し、選択された空き領域を分割して割り当てる実領域 1 3 2 を切り出す。まず実領域切出部 2 1 4 は、`size` が領域サイズ上限値 2 2 5 以上か否か判定する（ステップ 2 1 0 9）。`size` が領域サイズ上限値 2 2 5 以上の場合（ステップ 2 1 0 9 Y）には、実領域切出部 2 1 4 は、選択した空き領域から領域サイズ上限値 2 2 5 の整数倍であって `size` を越えない最大の領域を切り出す（ステップ 2 1 1 0）。次に切り出した実領域 1 3 2 とそのサイズをリストに登録する（ステップ 2 1 1 1）。次にステップ 2 1 0 3 と同様の処理によって渡された連続空き領域群について残りの空き領域を選択する（ステップ 2 1 1 2）。次に `size` からステップ 2 1 1 1 で登録した実領域 1 3 2 のサイズを差し引く（ステップ 2 1 1 3）。ステップ 2 1 1 2 において、残りの空き領域を選択することにより、今回のステップ 2

1 1 0 で切出した実領域 1 3 2 と以降で実行するステップ 2 1 0 5、ステップ 2 1 1 5 またはステップ 2 1 1 0 で切り出す空き領域とは連続することになる。s i z e が 0 になった場合（ステップ 2 1 1 4 Y）には、ステップ 2 1 1 7 へ進む。それ以外の場合は、ステップ 2 1 0 4 に戻る。

【 0 0 3 7 】

s i z e が領域サイズ上限値 2 2 5 未満の場合（ステップ 2 1 0 9 N）には、実領域切出部 2 1 4 は、B. A. で領域を切り出す（2 1 1 5）。すなわち s i z e を s i z e 以上の最小の 2 のべき乗数のサイズに切り上げる。そして選択された空き領域からこの切り上げたサイズの領域を切り出し、残りの領域があればそれを空き領域とする。次に実領域切出部 2 1 4 は、切り出した実領域 1 3 2 とそのサイズをリストに登録する（ステップ 2 1 1 6）。最後に実領域切出部 2 1 4 は、リストを割当処理部 2 1 3 に返して（ステップ 2 1 1 7）、処理を終了する。

【 0 0 3 8 】

上記実施形態の割当処理部 2 1 3 及び領域切出部 2 1 4 によれば、実領域の割当に際して、空き容量の大きな順に空き領域を選択しかつ連続する空き領域を選択することによって、割当済領域のフラグメンテーションと空き領域のフラグメンテーションを回避でき、発明が解決しようとする 4 つの課題のうちの第 1 の課題を解決することができる。また空き領域から割り当てる実領域を切り出していく、残りサイズが領域サイズ上限値未満となったときに始めて B. A. を適用するので、領域の使用効率を向上させることができ、第 2 の課題を解決することができる。また隣接する割当済領域が解放しにくいような空き領域を選択し、解放されにくい割当済領域に隣接して実領域を割り当てることによって、連続して解放され空き領域を拡大させる可能性が大きくなり、第 3 の課題を解決することができる。また分割数 3 1 2 に応じて仮想ボリューム 1 0 0 を負荷分散可能な領域に分割することができるので仮想ボリューム 1 0 0 への負荷を分散させることができ、第 4 の課題を解決することができる。特に第 1 の実施例では、ステップ 2 1 1 0 で選択した空き領域から領域サイズ上限値 2 2 5 の整数倍であって残りサイズを越えない最大の領域を切り出すので、空き領域の残りが領域サイズ上限値

2 2 5 未満になる可能性が生じ、領域の使用効率を向上させる第 2 の課題解決に寄与する。

【 0 0 3 9 】

図 6 は、第 2 の実施例において実領域切出部 2 1 4 の処理の流れを示すフローチャートである。第 2 の実施例は、第 1 の実施例の実領域切出部 2 1 4 のステップ 2 1 1 0 の代わりにステップ 2 1 2 0 を実行する。ステップ 2 1 2 0 では、実領域切出部 2 1 4 は、size を越えない最大の 2 のべき乗数のサイズをもつ実領域 1 3 2 を切り出す。

【 0 0 4 0 】

第 2 の実施例によれば、仮想ボリューム 1 0 0 に割り当てられる実領域は、領域サイズ上限値 2 2 5 を越える 2 べき乗数のサイズの複数の実領域 1 3 2 に分割され、一見すると割当済領域のフラグメンテーションが生じるように見える。しかし、ステップ 2 1 1 2 により切り出した実領域 1 3 2 の隣の空き領域が必ず選択されるので、切り出された複数の領域は連続することになり、フラグメンテーションを回避できる。なお第 2 の実施例の場合も上述したように発明が解決しようとする 4 つの課題を解決していることが理解される。

【 0 0 4 1 】

割当処理部 2 1 3 の処理が終了すると、仮想ボリューム定義部 2 1 1 は、新規作成又は更新された仮想ボリューム管理テーブル 2 2 1 の内容を管理コンソール 1 4 に表示する。システム管理者がこの表示内容を確認する。管理コンソール 1 4 を介して修正の入力があれば、仮想ボリューム定義部 2 1 1 は、この入力情報に基づいて当該仮想ボリューム管理テーブル 2 2 1 の実領域割当を修正する。

【 0 0 4 2 】

以上の確認及び修正が済んだ後に、仮想ボリューム定義部 2 1 1 は、仮割当て実領域リスト 3 1 6 内の実領域 ID 3 1 7 を実領域 1 3 2 のアドレス順にソートし、その結果の実領域 ID 3 1 7 群を割当済実領域リスト 3 1 5 の末尾に移動する。次に当該仮想ボリューム 1 0 0 の新規作成又は更新に伴って対応するアクセス変換テーブル 2 2 4 を作成又は更新し、処理を終了する。

【 0 0 4 3 】

仮割当て実領域リスト 3 1 6 内の実領域 I D 3 1 7 をソートするのは、割当処理部 2 1 3 の処理と協調して仮想ボリューム 1 0 0 に割り当てた実領域 1 3 2 をなるべく連続させるためである。このソート処理により実領域 1 3 2 が連続すれば、対応付ける仮想ボリューム 1 0 0 上のアドレス範囲とストレージ装置 1 3 上の実領域 1 3 2 のアドレス範囲の組の数を減少させることができ、アドレス変換テーブル 2 2 4 のサイズを小さくすることができる。すなわち本発明が解決しようとする課題のうちの第 1 の課題の解決に寄与できる。

【 0 0 4 4 】

図 7 は、本実施形態において仮想ボリュームに実領域を割り当てるときの原理を説明する図である。図 7 (a) は、ストレージ群 A , B , C があり、その転送性能が各々 8 0 M B / s , 6 0 M B / s , 4 0 M B / s とする。また空き容量の合計が各々 8 0 G B , 2 0 0 G B , 3 0 0 G B とする。割当処理部 2 1 3 は、要求性能を満足する空き容量の合計が最も大きいストレージ群を選択する。例えば 5 0 M B / s 以上の転送性能をもつ 6 0 G B の領域が要求されたとき、割当処理部 2 1 3 は、ストレージ群 B を選択する。またストレージ群 B の空き容量の中でできるだけ連続する大きい空き領域を選択する。これによって記憶領域のフラグメンテーションの回避が可能となる。

【 0 0 4 5 】

図 7 (b) は、あるストレージ群上の連続領域において、解放しやすさが 0 . 5 と 1 0 . 0 の割当済領域の間に空き領域 A が存在し、解放しやすさが 1 0 . 0 と 2 0 . 0 の割当済領域の間に空き領域 B が存在している状態を示している。このとき割当処理部 2 1 3 は、解放のしやすさが最も小さい割当領域に隣接する空き領域 A を選択する。また実領域切出部 2 1 4 は、解放のしやすさが最も小さい割当領域に連続するように実領域を割り当てる。解放しやすさが 1 0 . 0 及び 2 0 . 0 の割当済領域が将来解放される可能性が高く、従って連続空き領域が拡大する可能性が大きいためである。

【 0 0 4 6 】

図 7 (c) は、第 1 の実施例の効果を説明する図である。例えば 2 5 6 G B 以上の連続空き領域があるとし、この領域に要求サイズが 1 5 0 G B の実領域を割

り当てるものとする。領域サイズ上限値 2 2 5 は 3 2 G B とする。従来方式により、1 5 0 G B 以上で最小の 2 のべき乗数のサイズの領域を割り当てるとすると、2 5 6 G B の実領域を割り当てることになる。第 1 の実施例の実領域切出部 2 1 4 によれば、領域サイズ上限値 2 2 5 の整数倍で実領域を割り当てのために 1 6 0 G B の実領域の割当てで済み、従来方式に比べて $3 2 \text{ G B} \times 3 = 9 6 \text{ G B}$ の無駄な割当てを回避できる。より詳細に言えば、1 回目の割当てで 1 5 0 G B を越えない $3 2 \text{ G B} \times 4$ の領域を割り当て、2 回目の割り当てで残り 2 2 G B について B . A . を適用するので $3 2 \text{ G B}$ の領域を割り当て、結局 $3 2 \text{ G B} \times 5 = 1 6 0 \text{ G B}$ の実領域を割り当てることになる。また 1 6 0 G B の領域は解放しにくい割り当て済領域に連続するように割り当てられるので、図の空き領域の左端又は右端に寄せて割り当てられる。また第 1 の実施例で無駄になる領域は常に領域サイズ上限値 2 2 5 未満であることが理解される。

【 0 0 4 7 】

第 2 の実施例では、図 7 (c) について第 1 の実施例と割当て方式が異なるだけである。第 2 の実施例ではまず 1 5 0 G B を越えない最大の 2 のべき乗数のサイズの $3 2 \text{ G B} \times 4$ の領域を割り当てると、残りサイズが 2 2 G B となり、領域サイズ上限値 2 2 5 未満となり 2 回目の割当てで残り 2 2 G B について B . A . を適用するので $3 2 \text{ G B}$ の領域を割り当て、結局 $3 2 \text{ G B} \times 5 = 1 6 0 \text{ G B}$ の実領域を割り当てることになり、その効果は第 1 の実施例の場合と変わらない。第 2 の実施例の場合も無駄になる領域は常に領域サイズ上限値 2 2 5 未満となることが理解される。

【 0 0 4 8 】

図 8 を用いて本発明の第 3 の実施例について説明する。図 8 は、第 3 の実施例のシステム構成図である。図 8 の構成と図 1 の構成との違いは、仮想化スイッチ 1 1 がストレージ装置 8 1 3 に内蔵されていることである。すなわち仮想化スイッチ 1 1 とストレージ装置 8 1 3 は同じ筐体に格納されている。ストレージ装置 8 1 3 は、自身の L U を仮想化ボリューム 1 0 0 としてホストプロセッサ 1 2 がアクセスすることを可能にする。またストレージ装置 8 1 3 の仮想化スイッチ 1 1 には、他のストレージ装置 1 3 も接続可能である。仮想化スイッチ 1 0 0 は、

上記第 1 または第 2 の実施例と同様の機能をもつので説明を省略する。第 3 の実施例によれば、仮想化機能と仮想ボリューム割当機能をもつストレージ装置 8 1 3 は、第 1 または第 2 の実施例と同様に本発明が解決しようとする課題を解決することができる。

【 0 0 4 9 】

図 9 及び図 1 0 を用いて本発明の第 4 の実施例について説明する。図 9 は、第 4 の実施例のシステム構成図である。本実施例では、ストレージ装置 1 3 及び管理コンソール 1 4 はホストプロセッサ 1 2 に接続される。図 9 の構成と図 1 の構成との違いは、ホストプロセッサ 1 2 の処理部 1 2 1 および記憶部 1 2 2 を明示したことであり、処理部 1 2 1 および記憶部 1 2 2 は、そのプログラムを実行することによって仮想化スイッチ 1 1 が持つ仮想化機能を実現する。

【 0 0 5 0 】

図 1 0 は、第 4 の実施例における記憶部 1 2 2 内のプログラム及びデータの構成を示している。図 1 0 の記憶部 1 2 2 は、図 3 の記憶部 2 2 0 と異なり、仮想ボリューム定義プログラム 3 8 1、アクセス変換プログラム 3 8 2、割当処理プログラム 3 8 3、実領域切出プログラム 3 8 4 およびプール作成プログラム 3 8 5 を格納する。アクセス変換プログラム 3 8 2 は、処理部 1 2 1 によって記憶部 1 2 2 にロードして実行され、実行されるアプリケーションプログラムまたはファイルシステム制御プログラム（図示せず）が発行する仮想ボリューム 1 0 0 への入出力要求をストレージ装置 1 3 の LU 1 3 1 への入出力要求に変換する。

【 0 0 5 1 】

仮想ボリューム定義プログラム 3 8 1、割当処理プログラム 3 8 3、実領域切出プログラム 3 8 4 およびプール作成プログラム 3 8 5 は、処理部 1 2 1 によって記憶部 1 2 2 にロードされて実行され、それぞれ図 3 に示す仮想ボリューム定義部 2 1 1、割当処理部 2 1 3、実領域切出部 2 1 4 およびプール作成部 2 1 5 が持つ機能を実現する。その他の構成及び機能は、上記第 1 または第 2 の実施例と同様なので説明を省略する。第 4 の実施例によれば、ホストプロセッサ 1 2 の上記プログラム実行によってストレージ装置 1 3 の仮想化機能と仮想ボリューム割当機能を実現でき、第 1 または第 2 の実施例と同様に本発明が解決しようとする

る課題を解決することができる。

【 0 0 5 2 】

【発明の効果】

以上述べたように本発明によれば、記憶領域のフラグメンテーションを回避することができる。また記憶領域の使用効率を向上させることができる。また連続する空き領域のサイズを拡大することが可能となる。さらに負荷分散を考慮する記憶領域の割当が可能である。

【図面の簡単な説明】

【図 1】

実施形態のシステム構成図である。

【図 2】

実施形態の仮想化スイッチ 1 1 の内部構成を示す図である。

【図 3】

実施形態の処理部 2 1 0 および記憶部 2 2 0 の内部構成を示す図である。

【図 4】

実施形態の割当処理部 2 1 3 の処理の流れを示すフローチャートである。

【図 5】

第 1 の実施例の実領域切出部 2 1 4 の処理の流れを示すフローチャートである。

【図 6】

第 2 の実施例の実領域切出部 2 1 4 の処理の流れを示すフローチャートである。

【図 7】

実施形態の実領域割当の原理を説明する図である。

【図 8】

第 3 の実施例のシステム構成図である。

【図 9】

第 4 の実施例のシステム構成図である。

【図 1 0】

第 4 の実施例の記憶部 1 2 2 の内部構成を示す図である。

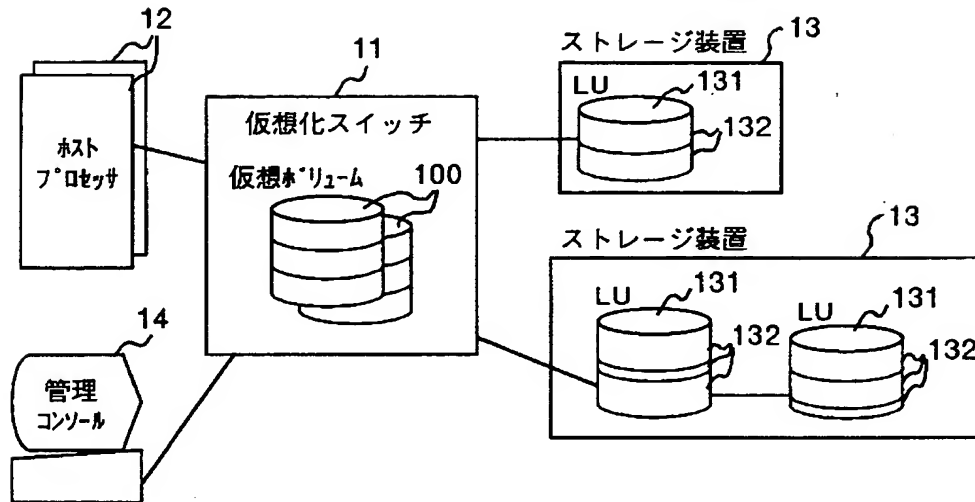
【符号の説明】

1 1 : 仮想化スイッチ、1 2 : ホストプロセッサ、1 3 : ストレージ装置、1
3 1 : LU、1 3 2 : 実領域、2 1 1 : 仮想ボリューム定義部、2 1 3 : 割当処
理部、2 1 4 : 実領域切出部、2 2 1 : 仮想ボリューム管理テーブル、2 2 2 :
実領域管理テーブル、2 2 3 : プール管理テーブル、2 2 5 : 領域サイズ上限値

【書類名】図面

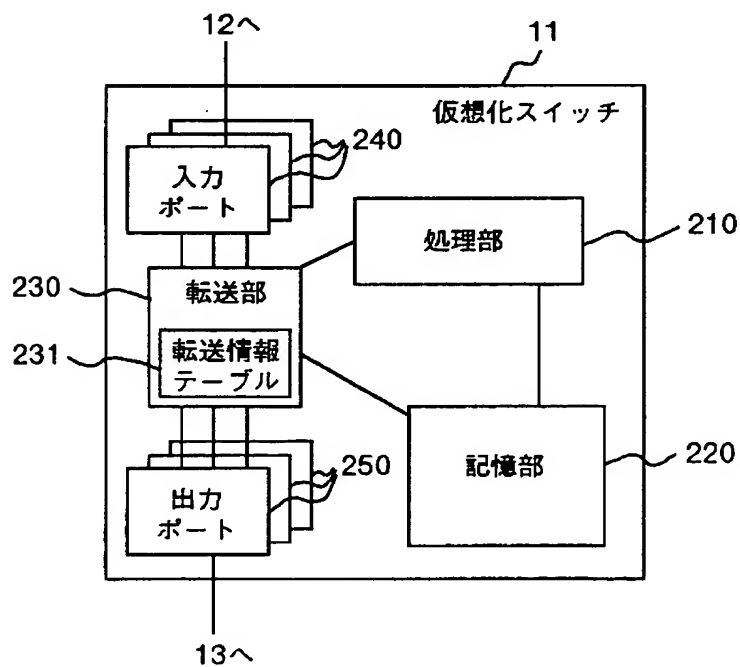
【図 1】

図 1



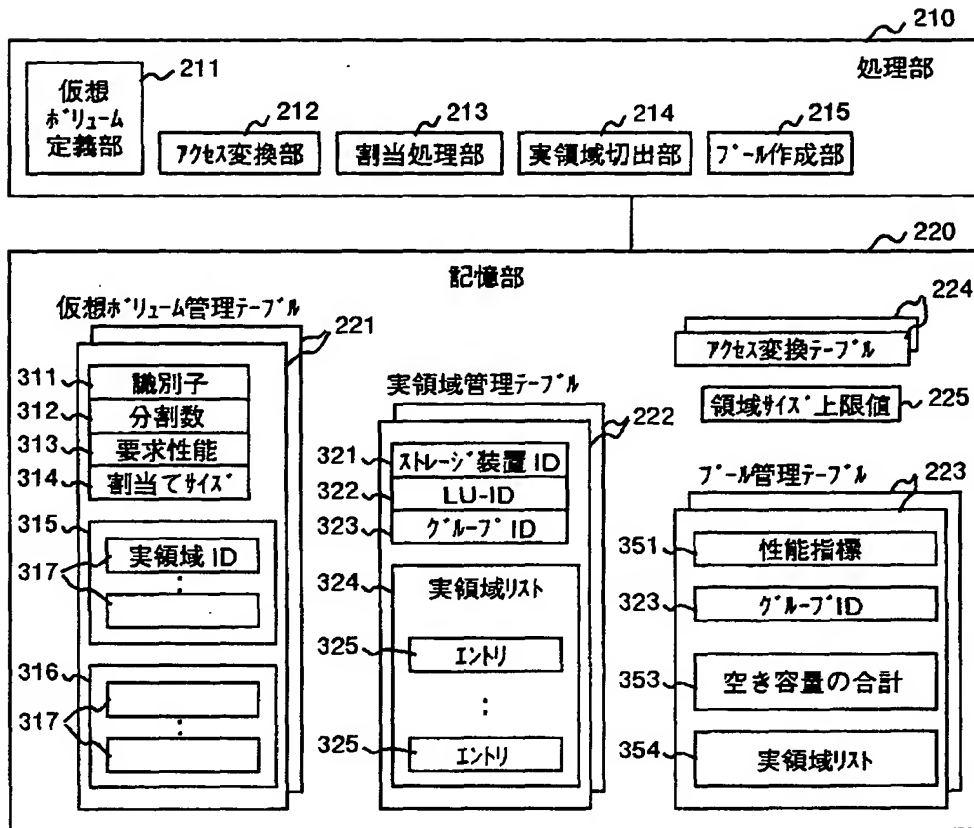
【図 2】

図 2



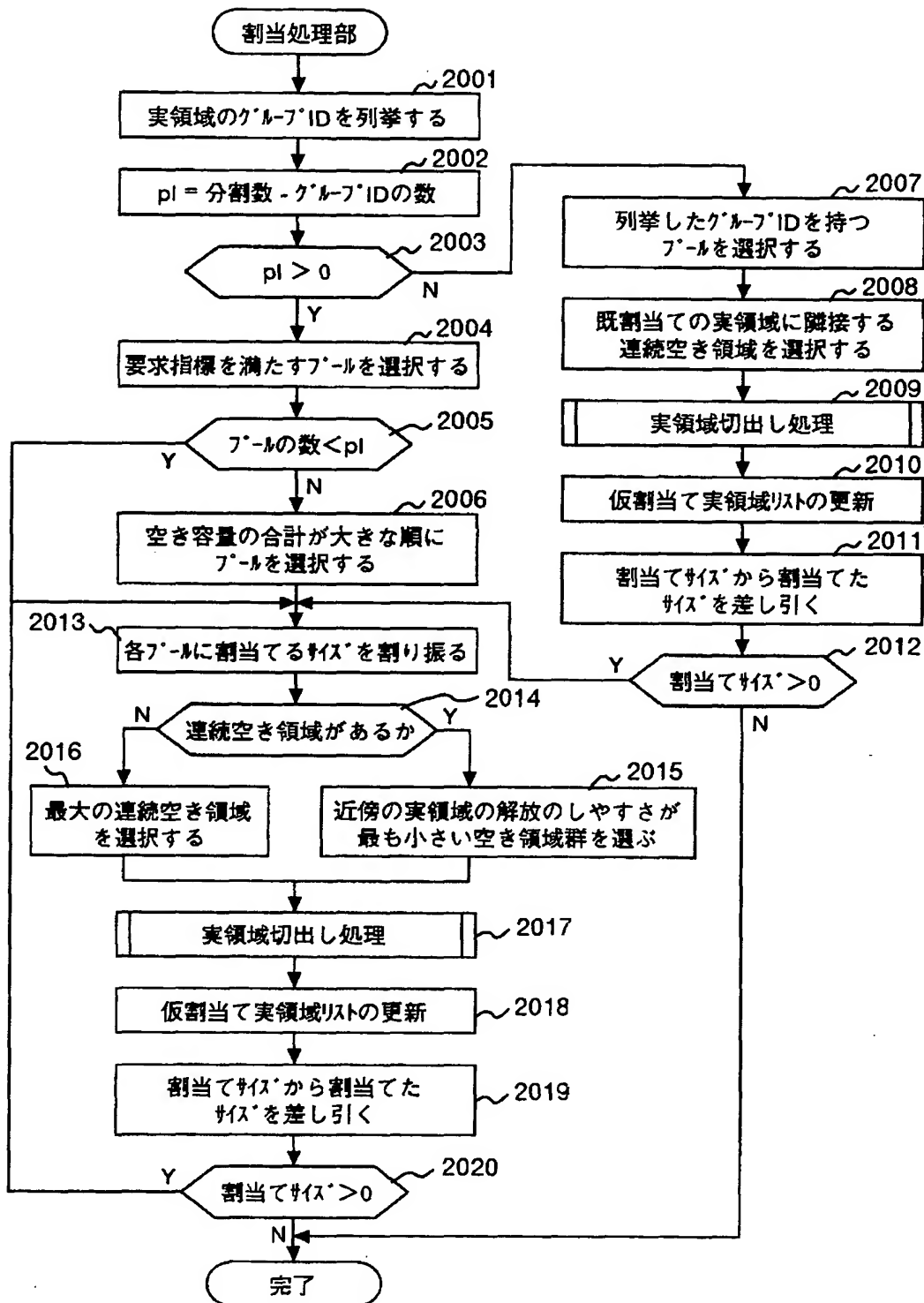
【図 3】

図 3



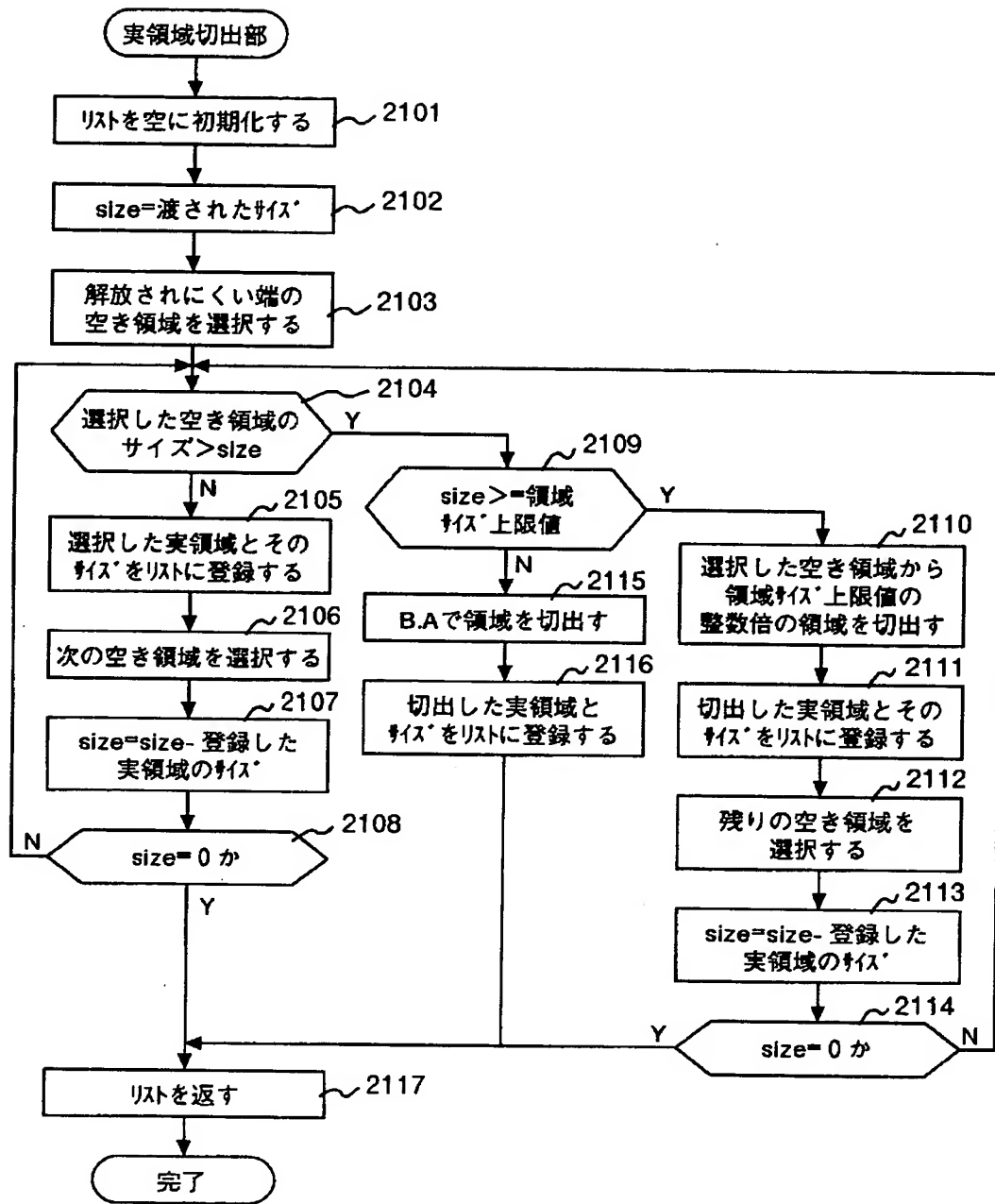
【図 4】

図 4



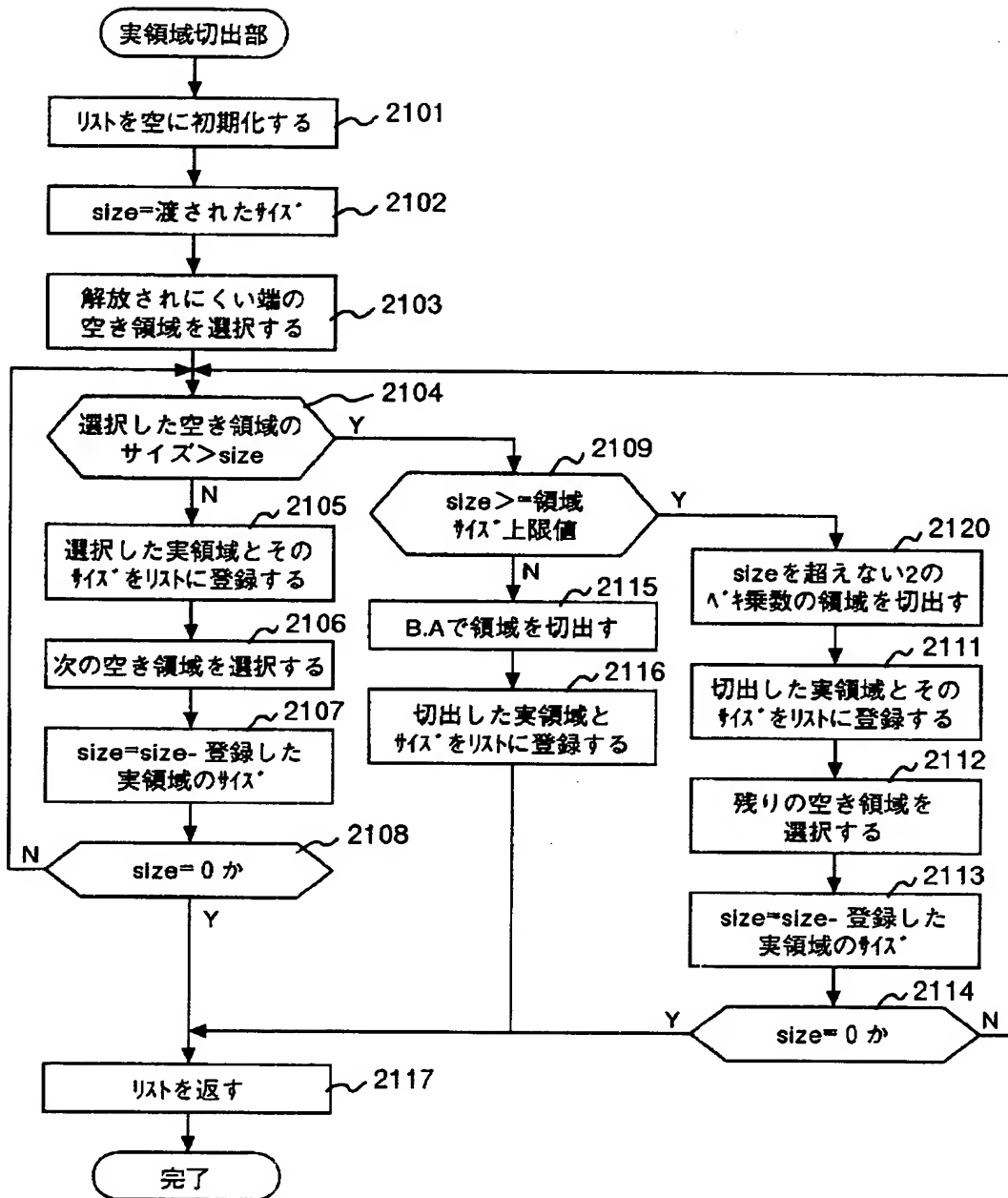
【図 5】

図 5



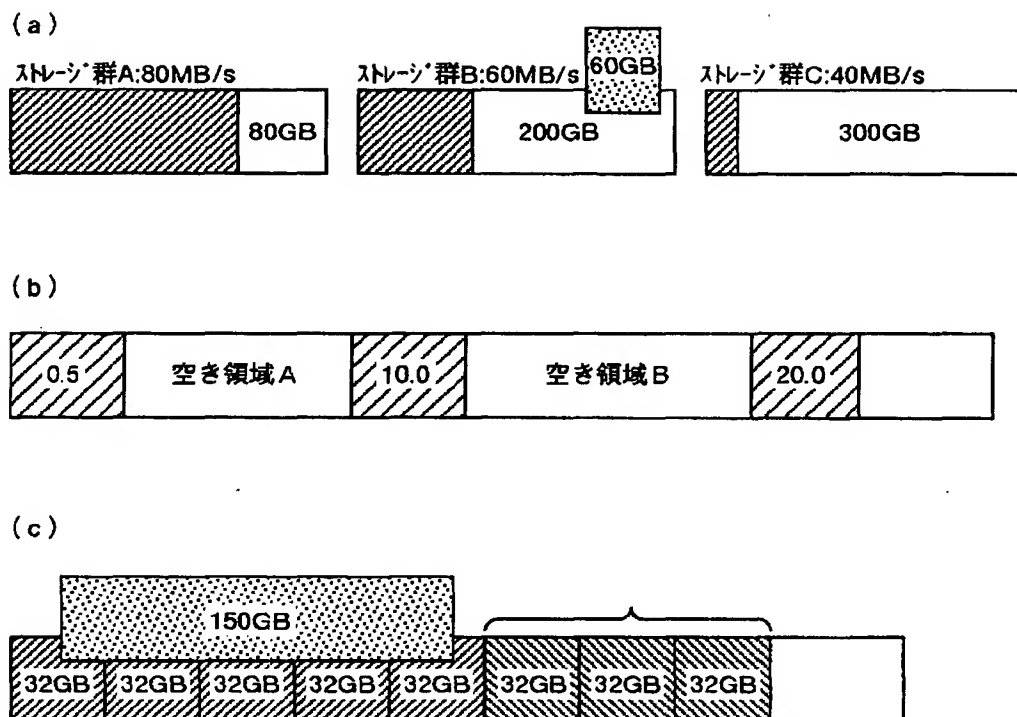
【図 6】

図 6



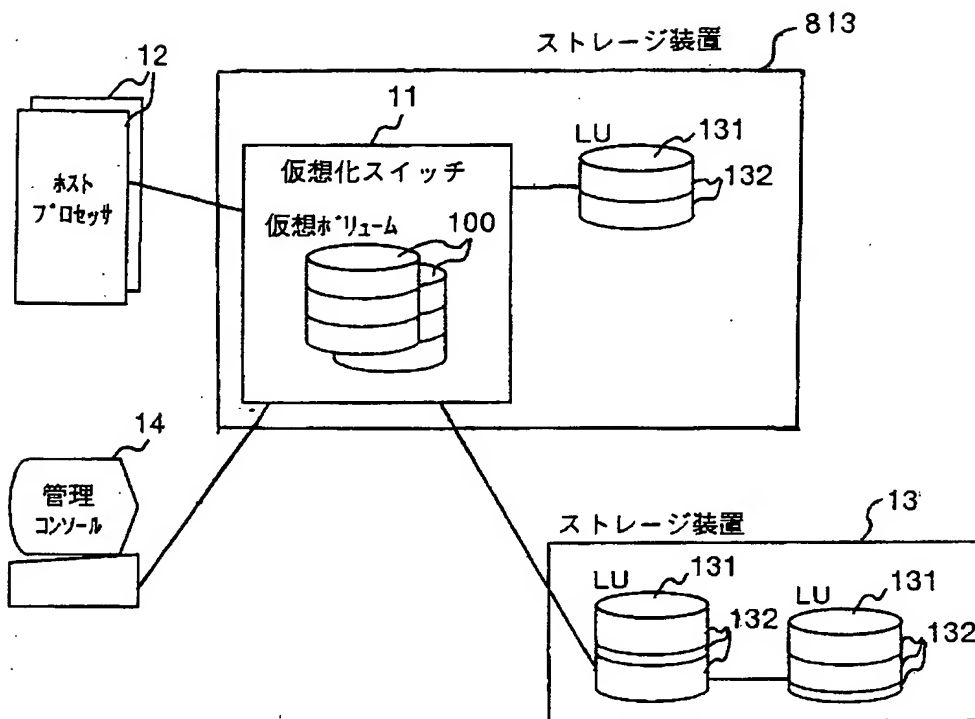
【図 7】

図 7



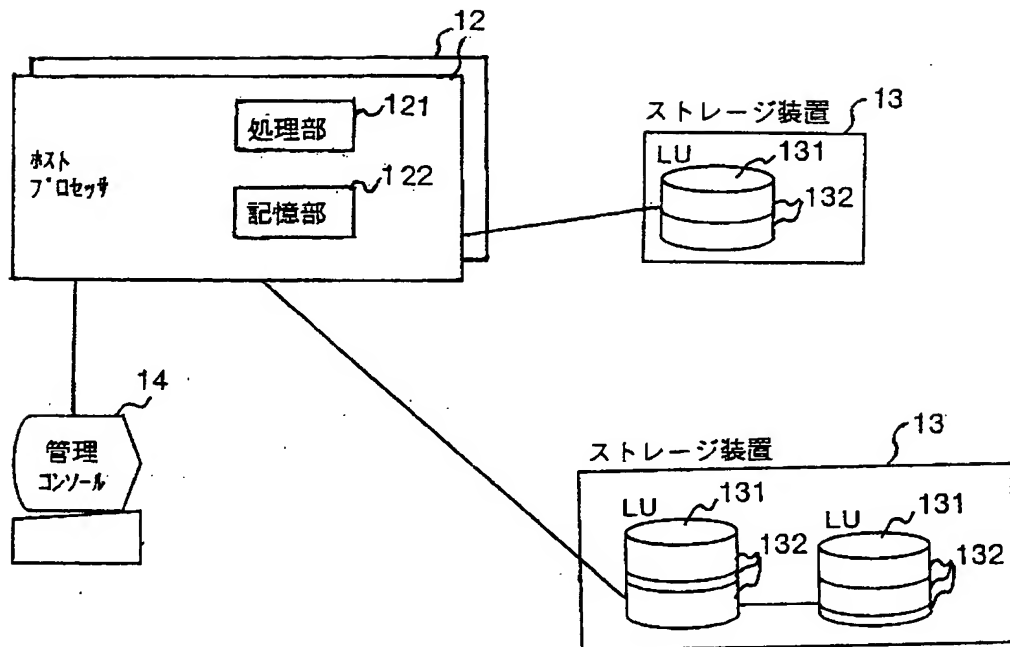
【図 8】

図 8



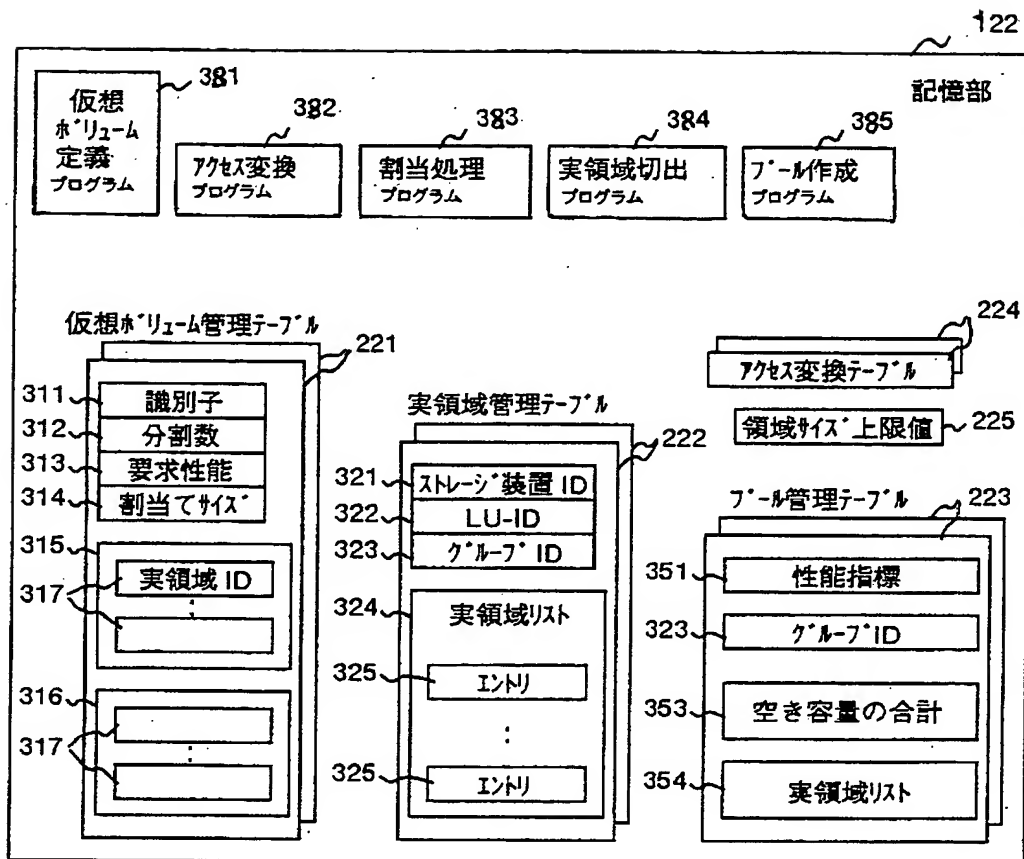
【図9】

図 9



【図10】

図 10



【書類名】 要約書

【要約】

【課題】 仮想ボリュームに対して少なくとも1台のストレージ装置の空き記憶領域を割り当てる技術において、フラグメンテーションを回避し、また記憶領域の使用効率を向上させる実領域割当を行う。

【解決手段】 仮想化スイッチ11は、管理コンソール14から指定された要求サイズに対して残りサイズが指定された領域サイズ上限値未満となるまで空きの記憶領域から記憶領域を切り出して割り当て、残りサイズが領域サイズ上限値未満となったとき、この残りサイズについて残りサイズ以上の最小の2のべき乗数のサイズの記憶領域を空きの記憶領域から切り出して割り当てる。また仮想化スイッチ11は、空きの記憶領域が複数の連続空き領域から成るとき、最大サイズをもつ連続空き領域を割当対象として選択する。

【選択図】 図1

出 願 人 履 歴 情 報

識別番号 〔000005108〕

1. 変更年月日 1990年 8月31日

 [変更理由] 新規登録

 住 所 東京都千代田区神田駿河台4丁目6番地
 氏 名 株式会社日立製作所